Question 1: Write the source code of classes CriticalException and NonCriticalException. (10p)

```
public class CriticalException extends java.io.IOException {
   public CriticalException(String arg0) {
      super(arg0);
   }
}
public class NonCriticalException extends RuntimeException {
   public NonCriticalException(String arg0) {
      super(arg0);
   }
}
```

Question 2: Write the source code of classes Person and Employee. (15p)

```
public abstract class Person implements java.io.Serializable {
   private String name, address, phoneNr;
   private int nationalID;

   public Person(String name, int nationalID) {
      this.name = name; this.nationalID = nationalID;
   }

   public void setAddress(String address) { this.address = address; }
   public void setPhoneNr(String phoneNr) { this.phoneNr = phoneNr; }

   public String getAddress() { return address; }
   public String getPhoneNr() { return phoneNr; }
   public String getName() { return name; }
   public int getNationalID() { return nationalID; }

   public String toString() {
      String str = "Name: " + name + ", National ID: " + nationalID;
      return str;
   }
}

public class Employee extends Person {
   private static final long serialVersionUID = 1L;
   private final int empID;
   private Date enlisted;

   public Employee(String name, int nationalID, int empID) {
      super(name, nationalID);
      this.empID = empID;
      enlisted = new Date();
   }
   public int getEmpID() { return empID; }
   public Date getEnlisted() { return enlisted; }
}
```

Question 3: Write the source code of class Room. The details of its methods are as follows: (25p)

```
import java.util.*;

public class Room implements java.io.Serializable{

   private static final long serialVersionUID = 1L;
   private final int roomNr;
   public final static int maxRoomCapacity = 10;
   private int capacity;
   private double dailyRate;
   private LinkedList<Guest> guests;
   private Visit v;

   public Room(int roomNr, int capacity, double dailyRate) throws CriticalException {
      this.roomNr = roomNr;
      setCapacity( capacity );  //or add try-catch
```

```
    setDailyRate( dailyRate );
    guests = new LinkedList<Guest>();
    v = new Visit(new Date(), this);
  }

  public int getRoomNr() { return roomNr; }
  public int getCapacity() { return capacity; }
  public int getGuestCount() { return guests.size(); }
  public double getDailyRate() { return dailyRate; }

  public void setCapacity( int newCapacity ) throws CriticalException {
    if( newCapacity <= 0 || newCapacity > maxRoomCapacity )
      throw new CriticalException("invalid capacity: " + newCapacity);
  }

  public void setDailyRate(double dailyRate) throws NonCriticalException {
    if( guests.size() == 0 )
      this.dailyRate = dailyRate;
    else throw new NonCriticalException
      ("Cannot change rate when room is already occupied");
  }

  public void addGuest(Guest guest) throws CriticalException {
    for( Guest existing : guests )
      if( existing == guest )
        throw new CriticalException("Duplicade guest: "+guest);
    guests.add(guest);
  }

  public Visit emptyRoom() {
    v.endVisit();
    guests = new LinkedList<Guest>();
    return v;
  }

  public Guest searchGuest(String name) {
    for( Guest existing : guests )
      if( existing.getName() == name )
        return existing;
    return null;
  }

  public String toString() {
    String str = "Room Nr: " + roomNr + "\nCapacity: " + capacity + "\nGuests: ";
    for( Guest existing : guests )
      str += "\n\t"+existing.toString();
    return str;
  }

  public String getGuestNames() {
    String str = new String();
    for( Guest existing : guests )
      str += existing.toString() + "\t";
    return str;

  }

  public Visit getVisit() { //to report current revenue
    return v;
  } //or add a method which calculates the revenue
}
```

Question 4: Write the source code of the classes ReportCurrentRevenue and ReportPastIncome. (25p)

```
import java.util.*;
public class ReportCurrentRevenue
implements Runnable {
  private Hashtable<Integer,Room> rooms;
  public ReportCurrentRevenue(
      Hashtable<Integer,Room> rooms ) {
      this.rooms = rooms; }
  public void run() {
    double amount = 0.0;
    for( Room room : rooms.values() ) {
      Visit v = room.getVisit();
      v.endVisit();
      amount += v.getTotalPrice();
    }
    System.out.println("Revenue:" +
    amount);
  }
}
```

```
public class ReportPastIncome implements
Runnable {
  private LinkedList<Visit> visits;
  public ReportPastIncome(
    LinkedList<Visit> visits) {
    this.visits = visits; }
  public void run() {
    double amount = 0.0;
    for( Visit v : visits ) {
      amount += v.getTotalPrice();
    }
    System.out.println("Past income:" +
    amount);
  }
}
```

Question 5: Write the source code of only the following methods of the class Hotel. (25p)

```
  public void report() {
    pool = Executors.newCachedThreadPool( );
    pool.execute( new ReportCurrentRevenue(rooms) );
    pool.execute( new ReportPastIncome(visits) );
    pool.shutdown( );
  }

  public void restore( String fileName ) {
    while( pool.isTerminated() );
    try {
      ObjectInputStream str = new ObjectInputStream(
           new FileInputStream(fileName));
      rooms = (Hashtable<Integer,Room>)str.readObject();
      visits = (LinkedList<Visit>)str.readObject();
      str.close();
    }
    catch(ClassNotFoundException e) { e.printStackTrace(); }
    catch(IOException e) { e.printStackTrace(); }
  }

  public int findRoomNrOfGuest( String name ) {
    for( Room room : rooms.values() ) {
      Guest guest = room.searchGuest(name);
      if( guest != null )
        return room.getRoomNr();
    }
    return 0;
  }
```